Assignment 4: Data Viz for Advocacy

Ivan Aristy — iae225 *CS-GY 6313 B*

1 Interactive Visualization

1.1 Question

How are men's mental states in the US?

The question is a bit general, but I want to communicate to a general audience that is not aware of the high levels of mental health issues that men face in the US.

1.2 Data

1.2.1 Data Source

There are a few data sources to get information from.

The CDC holds lots of information, but particularly I looked into:

1. Behavioral Risk Factor Surveillance System (BRFSS) Survey
2. Household Pulse Survey
3. National Health and Nutrition Examination Survey (NHANES)
4. National Health Interview Survey (NHIS)

However, much to my dismay, lot's of the Data I was looking for was not available. I was able to retrieve information for suicide rates, but the data regarding mental health for men was quite limited.

I ended up using `https://www.cdc.gov/suicide/facts/data.html#cdc_data_surveillance_section_4-suicide-rates` for suicide rate information, and `https://www.apa.org/monitor/2015/12/numbers` for general information and stats regarding suicide rates.

1.2.2 Serving Information

For serving information, we use D3's csv function:

```
import * as d3 from "d3";
import { Dataset } from "@/types/types";
import theme from "@/types/themes";

export async function loadRatesData(): Promise<Dataset[]> {
  const data = await d3.csv("/rates.csv", d3.autoType);
  console.log(data);

  return [
    {
      label: "Total␣Population",
      data: data.map((d: any) => ({ x: d["Year"],
       y: d["Total␣Population"] })),
      color: "#d4c2d4",
    },
    {
      label: "Male",
      data: data.map((d: any) => ({ x: d["Year"], y: d["Male"] })),
      color: "#cbd4c2",
    },
    {
      label: "Female",
      data: data.map((d: any) => ({ x: d["Year"], y: d["Female"] })),
      color: "#c2c2d4",
    },
  ];
}
```

and pass the data to the chart component through a wrapper.

1.3 Visualization

1.3.1 Frontend Setup

The frontend is a simple NextJS application that uses the `D3` library to create the chart, and react hooks to update and keep track of state.

My main secondary goal for this visualization was to make the design look sleek and simple.

As opposed to my previous visualization, I wanted to make a website that was more visually appealing and easier to understand, as well as having goodies like smooth transitions and a responsive design.

For smooth transitions I used the Lenis library, which allowed for smooth scrolling. This greatly improved the feel of the website, and allowed me to dynamically render the chart when the user scrolls to the chart section.

```
const [isVisible, setIsVisible] = useState(false);
    const ref = useRef<HTMLDivElement>(null);

    // Lenis hook to listen for scroll events
    useLenis(() => {
      if (ref.current && !isVisible) {
        const { top, bottom } = ref.current.getBoundingClientRect();
        const windowHeight = window.innerHeight;

        if (top + 300 < windowHeight && bottom > 0) {
          setIsVisible(true);
        }
      }
    });
```

1.3.2 Components

To allow for interactivity, we create a few drop down labels that allow the user to select the data that they want to plot. For example, here is the code for the "Champion" Label:

```
  <label>
    Champion:
    <select
      value={selectedChampion}
      onChange={(e) => setSelectedChampion(e.target.value)}
    >
      {champions.map((champ) => (
        <option key={champ} value={champ}>
          {champ}
        </option>
      ))}
    </select>
  </label>
```

This also allows you to type out the champion name instead of going through every single champion in the dropdown.

Additionally, I implemented another QoL feature that deals with situations where the user selects a patch range that is not valid. For example, if the user sets start patch to 14.20 and end patch is at 14.10, the code will automatically adjust the end patch to 14.20.

1.3.3 Visualization Logic

I revisited D3 to create a custom chart, building upon my previous experience with the library.

Working directly with D3 again proved to be more challenging but also rewarding, as it deepened my understanding of chart components and their implementation.

I gained a more explicit appreciation for the building blocks of a chart:

- Margins: Carefully planned margins to ensure sufficient space for elements like axes, labels, and legends.
- Scales: Mapped dynamically to the data, ensuring that the chart is responsive to changes in data.
- Line: This time, I used 3 lines to represent the data, each with a different color.
- Tooltip:
- Circles: Since we were using 3 lines and transitions, the circles
- Dynamics: I progresively drew the lines, and added a transition to make the chart more visually appealing. I also believe that this draws attention to the chart, as the user is more likely to notice the chart if it is moving. Plus it highlights how rates increase over time.
- Axes: Created axes, gridlines, and basic labels for intuitive and fast understanding by the user.

1.4 Improvements

1.4.1 Champion Image

A major piece of feedback was to include the champion image in the visualization. This would allow the user to quickly identify the champion they are looking at, not having to rely on the champion name alone, which could be pretty small on some devices.

This was actually added.

1.4.2 Small Patch Ranges

We have a degenerate case where the user selects a patch range that is so small that visualizing it in a line chart is not very useful.

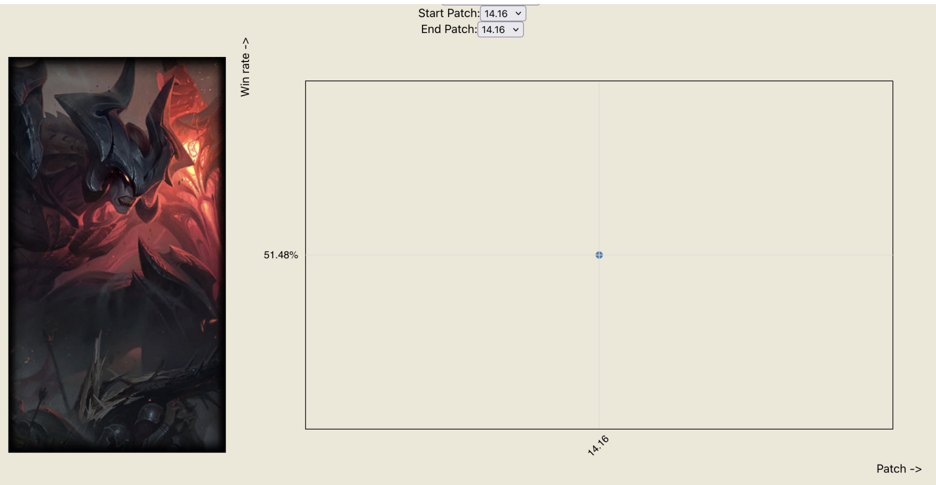
See Figure 1 and Figure 2 for an example of this.

I did not create a fix for this specific situation, since the Visualization is for users to see how their champion has been impacted over time.

1.4.3 Lane Specific Data

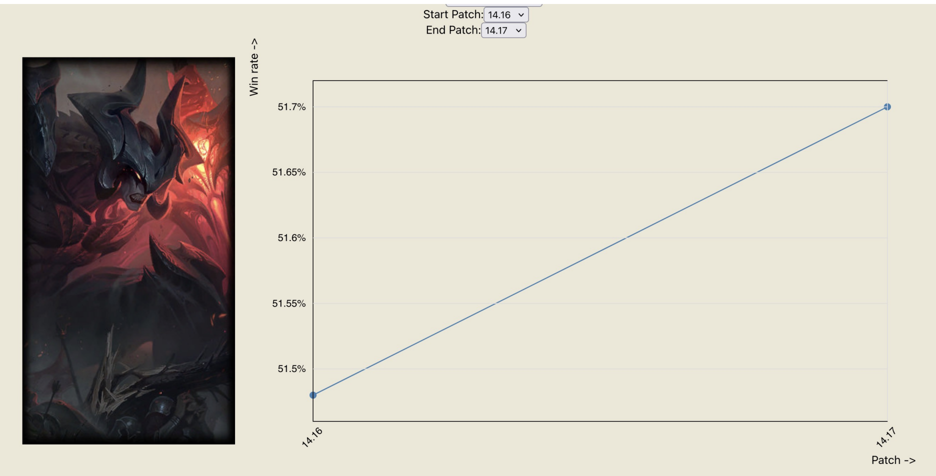
Another piece of feedback was to include lane specific data.

Why? Well, a botlane champion will have a different winrate if they are played toplane. In layman's terms, the best pizza is not a very good burger. Also, some champions are played in multiple lanes, so it would be interesting to see how their winrate changes depending on the lane.



Start Patch: 14.16
End Patch: 14.16

Figure 1: Solo patch range.



Start Patch: 14.16
End Patch: 14.17

Figure 2: Size 2 patch range.

I deemed this to be beyond the scope of the assignment, but it is a good idea.

1.5 Conclusion: Do we answer the question?

I believe for both: